Short Answer:

Answer the following questions with complete sentences in your own words. You are encouraged to conduct your own research online or through other methods before answering the questions. If you research online, please consult multiple sources before you write down your answers. You are expected to be able to explain your answers in detail (Provide examples to each question).

1. What are the features provided by ASP.NET Core
2. What are the advantages of ASP.NET Core over ASP.NET?
3. Where could we add services required by the application in ASP.NET Core MVC?
4. Where could we configure the request pipeline by adding middleware and defining how the application will respond to HTTP requests?
5. What is Kestral?
6. What is middleware?
7. How to read the configuration from appsettings.json in your application?
8. What's the JSON file that sets the launching config?
9. How can we inject the service dependency into the controller?
10. Naming the service life?

Coding Questions:

Write code in C#/.NET Core MVC to solve following problems. Please write your own answers. You are highly encouraged to present more than one way to answer the questions. Please follow best practice when you write the code so that it would be easily readable, maintainable, and efficient. Clearly state your assumptions if you have any. You may discuss with others on the questions, but please write your own code.

Develop an **ASP.NET Core MVC Application** that uses ADO.NET to communicate with database.

Tables:

- Student(id, firstname, lastname, email)
- Student_Course (id, studentlid, courseId)
- Course(id, name, description, professorId)
- Professor(id, firstname, lastname, email, office, title)

【Relationships:

- A student can take any number of courses. A course can be taken by different students.
- Each course is only taught by 1 professor. A professor can teach more than 1 courses.
- No direct relationship between students and professors. 】

1. Onboard Models to ASP.NET Core MVC Application based on the table defined

2. Create a folder named DAO and a CourseDAO.cs within it which includes the following methods:

Hint: add connectionString to the appsettings.json and load the connectionString via configuration object

a. Add
1) AddStudent - this method takes a student object and saves it to the db
2) AddCourse - this method takes a course object and saves it to the db
3) AddProfessor - this method takes a professor object and saves it to the db
4) AssignStudentToCourse - this method takes a studentId and a courseId and saves the information to the db
5) AssignProfessorToCourse - this method takes two inputs: a prodessorId and a courseId, and update the information to the db

b. Update
1) UpdateStudent – this method takes a student object and updates it to the db
2) UpdateCourse - this method takes a course object and Updateit to the db
3) UpdateProfessor - this method takes a professor object and Updateit to the db

c. Data Retrieval
1) FindStudentCoursesByStudentEmail - this method takes an email string and returns a List of courses he/she takes
2) FindProfessorCoursesByName - this method takes a name string (including both firstname and lastname) and returns a List of courses he/she taught
3) FindCourseById - this method takes a courseId and returns a Course object along with the professor's information

3. Inject this CourseDAO.cs class to the HomeController and verify your behaviors there